

# NAVAL POSTGRADUATE SCHOOL Monterey, California



## ALLOCATION OF JOBS TO UNEQUALLY- CAPABLE PROCESSORS: A PLANNING APPROACH

by

Donald P. Gaver ✓  
Patricia A. Jacobs ✓  
Kevin Becker ✓  
Siriphong Lawphongpanich ✓

September 1996

Approved for public release; distribution is unlimited.

Prepared for: Naval Postgraduate School  
Monterey, CA 93943

FedEx  
Box 14/2  
DPS OF-90-010

NAVAL POSTGRADUATE SCHOOL  
MONTEREY, CA 93943-5000

Rear Admiral M. J. Evans  
Superintendent

Richard Elster  
Provost

This report was prepared and funded by the Naval Postgraduate School.

Reproduction of all or part of this report is authorized.

This report was prepared by:



# REPORT DOCUMENTATION PAGE

*Form Approved*  
*OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE <p style="text-align: center;">September 1996</p>	3. REPORT TYPE AND DATES COVERED <p style="text-align: center;">Technical</p>
----------------------------------	---	--

4. TITLE AND SUBTITLE  Allocation of Jobs to Unequally-Capable Processors: A Planning Approach	5. FUNDING NUMBERS  N/A
--	-------------------------------

6. AUTHOR(S)  Donald P. Gaver, Patricia A. Jacobs, Kevin Becker, and Siriphong Lawphongpanich	
---	--

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  Naval Postgraduate School Monterey, CA 93943	8. PERFORMING ORGANIZATION REPORT NUMBER  NPS-OR-96-010
---	---

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)  N/A	10. SPONSORING / MONITORING AGENCY REPORT NUMBER
--	--

11. SUPPLEMENTARY NOTES
-------------------------

12a. DISTRIBUTION / AVAILABILITY STATEMENT  Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE
---	------------------------

13. ABSTRACT ( <i>Maximum 200 words</i> )  This paper addresses the problem in which jobs of different types arrive at a system that consists of a collection of individual and somewhat diverse processors. The processors differ in that each may specialize in one job type, but may also do others. Job types that are totally incompatible with a processor have an infinite service on that processor, but degrees of incompatibility may exist, and are modeled here. Using static queuing models, several practical performance measures may be evaluated, and optimal allocation of jobs to processors are obtained by solving linear and nonlinear programming problems. To illustrate, several numerical examples are provided. It is shown that jobs are not always most advantageously assigned to their most expert servers.
--

14. SUBJECT TERMS  queuing models; allocation of jobs to unequal servers	15. NUMBER OF PAGES <p style="text-align: center;">31</p>
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT <p style="text-align: center;">Unclassified</p>	18. SECURITY CLASSIFICATION OF THIS PAGE <p style="text-align: center;">Unclassified</p>	19. SECURITY CLASSIFICATION OF ABSTRACT <p style="text-align: center;">Unclassified</p>	20. LIMITATION OF ABSTRACT  <p style="text-align: center;">UL</p>
--	---	--	---



# ALLOCATION OF JOBS TO UNEQUALLY-CAPABLE PROCESSORS: A PLANNING APPROACH

D. P. Gaver

P. A. Jacobs

K. Becker

S. Lawphongpanich

## ABSTRACT

This paper addresses the problem in which jobs of different types arrive at a system that consists of a collection of individual and somewhat diverse processors. The processors differ in that each may specialize in one job type, but may also do others. Job types that are totally incompatible with a processor have an infinite service on that processor, but degrees of incompatibility may exist, and are modeled here. Using static queuing models, several practical performance measures may be evaluated, and optimal allocation of jobs to processors are obtained by solving linear and nonlinear programming problems. To illustrate, several numerical examples are provided. It is shown that jobs are not always most advantageously assigned to their most expert servers.

## 1. Problem Formulation

Consider a service system whose individual service facilities (servers) are *unequal* in their capacities to serve different job types. This means that, in some sense, a job of type  $j$  is most expeditiously done by a server of type  $i = i(j)$ , and quantifiably less so by other servers. It may well be that a good start is achieved by assigning jobs to those servers who require the least mean time to finish them,



but clearly this is not optimal if it tends to overload a few efficient servers and leaves others idle. A good dynamic approach might be to allocate the excess backlog of the efficient servers to others whenever that backlog is excessive. But this requires constant monitoring of queue lengths, or possibly current elapsed job service time. We do not investigate such dynamic rules here.

We propose to study a static queuing model that selects from the incoming job traffic stream of jobs of classes  $j = 1, 2, \dots, J$  a subset that is directed to each of the available servers: if a job is of type  $j$  it goes to server type  $i$  with probability  $a_{ji}$ . The procedure is based on classical M/G/1 queuing theory, and requires mathematical programming in order to optimize the allocation from the perspective of either servers or jobs (customers). The allocation can be both deterministic or randomized. In the deterministic case, all jobs of one type are assigned to a single server. For the randomized allocation, if a job of type  $j$  arrives, it can be assigned to server  $i$  with probability  $a_{ji}$ .

## 2. Model

The system studied is made up of  $I > 1$  single servers with different capabilities. The system is confronted by a Poisson (rate  $\lambda$ ) rate of demand of jobs, but of different job types: with independent probability  $p_j$  an arrival is a job of type  $j$ ,  $j \in (1, 2, \dots, J)$ . The different capabilities of servers to handle (serve) jobs of different types is reflected in their service times: the service time of job  $j$  on server  $i$ , denoted by  $S_{ji}$ , may tend to vary systematically with  $i$  for any of many possible reasons, one being each of training or recent experience by  $i$  with job type  $j$ . In fact, for some job types  $j'$  and some servers  $i'$ ,  $S_{j'i'}$  is effectively infinite if the servers in question have no capability to handle those particular jobs, so  $E[S_{j'i'}] = \infty$ ; this is a case of total incompatibility, and certainly exists in many practical settings. On the other hand it may be necessary to allocate jobs to



servers with which they are somewhat incompatible in order to avoid overloading the more compatible servers. It is to be expected that this phenomenon may tend to occur more extensively as traffic intensity increases, e.g. if  $\lambda$  increases, in which case a good planning policy choice of  $a_{ji}$  may well stave off disaster: the need to hastily add servers or reject jobs.

To investigate the effect of cross-assignment we introduce a static set of assignment probabilities  $a_{ji}$ . Thus if a type- $j$  job arrives in  $(t, t + dt)$  with probability  $\lambda p_j dt + o(dt)$ , it is assigned to server  $i$  with probability  $a_{ji}$ . We think of  $a_{ji}$  as a decision variable to be determined so as to optimize some measure of system performance. Total expected delay to all arriving jobs is one such measure, but the delays of some jobs may be more undesirable than those of others, in which case a total weighted expected delay can be studied. Note that, for job type  $j$ , we can select  $a_{ji}$  to be any real positive (actually non-negative) number such that  $\sum_{i=1}^I a_{ji} = 1$ , in which case a randomization device is needed to allocate an arrival of type  $j$  to its server, i.e.,  $a_{ji}$  represents a randomized job allocation. A more easily implemented approach would be to choose a single "best"  $i$ -value,  $i(j)$  for each  $j$ , and set

$$a_{ji} = \begin{cases} 1 & \text{if } i = i(j) \\ 0 & \text{otherwise.} \end{cases}$$

This approach is a deterministic allocation of jobs since it sends all jobs of type  $j$  to server  $i(j)$ . Later, it is shown that, although easier to implement, deterministic job allocation may not be desirable in practice.

Both deterministic and randomized job allocations provide a simple independent Poisson stream of jobs with independent service times to the individual servers. Moreover, if jobs are treated in arrival order (first-come, first-

served) at all servers, the stationary delay experienced at each of the  $I$  servers can be calculated using known results, e.g. the Pollaczek-Khintchine-Kendall formula. Parenthetically, static priority rules can also be followed, and formulas for long-run results are already available, but this extension is postponed.

Clearly the arrival rate at server  $i$  of all job types is

$$\lambda_i = \lambda \sum_{j=1}^I p_j a_{ji} = \lambda \bar{p}_i \quad i = 1, 2, \dots, I \quad (2.1)$$

and by standard results  $\lambda_i$  is the rate of a Poisson process independent of those prevailing for other servers ( $\neq i$ ). Now consider the effective service time,  $S_i$ , of jobs arriving at server  $i$ . A new job is of type  $j$  with probability  $p_j$ , and if it is dispatched to server  $i$  it experiences service time  $S_{ji}$ , so

$$S_i = S_{ji} \text{ with probability } \frac{p_j a_{ji}}{\bar{p}_i}. \quad (2.2)$$

Moments of  $S_i$  are calculated by conditioning, e.g.

$$E[S_i] = \sum_{j=1}^I E[S_{ji}] p_j a_{ji} / \bar{p}_i \quad (2.3)$$

$$E[S_i^2] = \sum_{j=1}^I E[S_{ji}^2] p_j a_{ji} / \bar{p}_i. \quad (2.4)$$

Also the Laplace transform is

$$E[e^{-\theta S_i}] = \sum_{j=1}^I E[e^{-\theta S_{ji}}] p_j a_{ji} / \bar{p}_i. \quad (2.5)$$

This latter becomes useful in case one wishes to use the probability of a long wait or delay as an optimization criterion; see Gaver and Jacobs (1988).

### 3. Performance Measures

Several service-system/queuing performance measures are relevant to overall system performance and can be formally evaluated. Then mathematical programming techniques can be applied to obtain optimal  $\{a_{ji}\}$  values.

#### 3.1 Traffic Intensity Parameters and Associated Optimization Problems

It has already been stated that  $\lambda_i = \lambda \bar{p}_i$ , and that  $E[S_i] = \sum_{j=1}^J p_j a_{ji} E[S_{ji}] / \bar{p}_i$ , so

the  $i^{\text{th}}$  server's traffic intensity is

$$\rho_i = \lambda_i E[S_i] = \lambda \bar{p}_i E[S_i] = \lambda \sum_{j=1}^J p_j a_{ji} E[S_{ji}]. \quad (3.1)$$

Using (3.1), several optimization problems can be formulated from the perspective of the servers. The first problem is to find a job allocation that minimizes a linearly weighted combination of the traffic intensities, i.e., solve the following optimization problem.

$$\text{P1:} \quad \min_{a_{ji}} \lambda \sum_{i=1}^I c_i \sum_{j=1}^J p_j a_{ji} E[S_{ji}]$$

$$\text{subject to} \quad \lambda \sum_{j=1}^J p_j a_{ji} E[S_{ji}] < 1 \quad \forall i = 1, \dots, I \quad (3.2)$$

$$\sum_{i=1}^I a_{ji} = 1 \quad \forall j = 1, \dots, J \quad (3.3)$$

$$a_{ji} \geq 0 \quad \forall i = 1, \dots, I; \quad j = 1, \dots, J \quad (3.4)$$

where the first constraint, (3.2), restricts the traffic intensity for each server  $i$  to be less than one in order to ensure the existence of long-run stationary delays. The second constraint ensures that the probability that job type  $j$  is assigned to server

$i$  sums to one. The parameter  $c_i$  in the objective function is the weight for the traffic intensity of server  $i$ . When  $c_i = 1$ , the objective function is equivalent to minimizing the average traffic intensity of the  $I$  servers.

When  $\lambda$  is sufficiently small, allocation  $a_{ji}$  satisfying constraints (3.3) and (3.4) may automatically satisfy constraint (3.2) and it may be eliminated. In this case, problem P1 has a simple solution. By interchanging the two summations, the objective function for P1 can be equivalently written as

$$\min \sum_{j=1}^J p_j \sum_{i=1}^I c_i a_{ij} E[S_{ji}].$$

For each job  $j$ , let  $\tilde{i}(j) = \arg \min \{c_i E[S_{ji}] : i = 1, \dots, I\}$  and allocate jobs as follows:

$$a_{ji}^* = \begin{cases} 1 & \text{if } i = \tilde{i}(j) \\ 0 & \text{otherwise} \end{cases} \quad \forall j = 1, \dots, J. \quad (3.5)$$

The allocation in (3.5) clearly satisfies (3.3) and (3.4). Moreover, it is also assumed to satisfy (3.4) when  $\lambda$  is sufficiently small. By construction,  $\tilde{i}(j)$  is the least cost server and, when  $c_i = 1$  for all  $i$ ,  $\tilde{i}(j)$  is the "best" or the most qualified server for job  $j$ . Since  $a_{ji}^*$  assigns job  $j$  to its least cost server, the corresponding objective function value must also be minimal. So the randomized and deterministic job allocations are the same when  $\lambda$  is sufficiently small.

When  $\lambda$  is too large, the allocation in (3.5) may not be feasible. In fact, there may not exist any feasible deterministic job allocation and the probabilistic job allocation may be the only choice. In which case, it is still optimal to assign the major proportion of job  $j$  to server  $\tilde{i}(j)$ . To minimize total cost, it is only necessary to divert just enough of jobs  $j$  to other servers to ensure that constraint (3.2) is satisfied. In practice, this may not be acceptable since server  $\tilde{i}(j)$  is likely

to have a traffic intensity (loading) close to one. To prevent this, several alternative optimization models for probabilistic job allocation are described below.

The first model minimizes the weighted sum of squared intensity of each server and can be stated as follows:

$$\text{P2:} \quad \min \sum_{i=1}^I c_i \left( \lambda \sum_{j=1}^J p_j a_{ji} E[S_{ji}] \right)^2$$

subject to constraints (3.2), (3.3), and (3.4).

The squared objective function of P2 penalizes servers with higher traffic intensity more than those with lower intensities. P2 can be solved as a quadratic programming problem (see, e.g., Bazarra, Sherali and Shetty, 1993). Since the objective function for P2 is convex, the solution is guaranteed to be globally optimal.

The second model tends to equalize the traffic intensity among all servers by minimizing the maximum intensity, i.e.,

$$\text{P3:} \quad \min \max_i \left\{ \lambda \sum_{j=1}^J p_j a_{ji} E[S_{ji}] \right\}$$

subject to constraints (3.2), (3.3), and (3.4).

The objective function for P3 is piecewise linear and convex. Thus, P3 is a nonlinear programming problem. However, it can be converted into a linear problem by introducing an auxiliary variable  $z$  and additional constraints to calculate the maximum traffic intensity as follows:



P4:  $\min z$   
subject to constraints (3.2),(3.3),(3.4), and

$$\lambda \sum_{j=1}^J p_j a_{ji} E[S_{ji}] \leq z \quad \forall i = 1, \dots, I \quad (3.6)$$

An optimal solution to P4 tends to assign jobs to servers so that their intensities all equal  $z^*$ , the optimal value of  $z$ .

Replacing constraint (3.4) with the following constraint translates randomized job allocation problems P2–P4 into deterministic ones.

$$a_{ji} \in \{0, 1\} \quad \forall i = 1, \dots, I; \quad j = 1, \dots, J. \quad (3.7)$$

Constraint (3.7) simply restricts  $a_{ji}$  to be either 0 or 1 and the resulting problems become integer programming problems, a more difficult class of problems to solve. However, if the optimal allocation differs from the allocation in (3.5), then some jobs may not be given to the most qualified servers and the optimal allocation may be hard to accept in practice. For the remainder, we focus on the randomized job allocation problems with a more operationally meaningful objective (penalty) function based on expected or mean job delay, or mean non-linearly-length-penalized job delay.

### 3.2 Optimization of Mean Functions of Total Job Delay

The Pollaczek-Khintchine-Kendall formula provides the expected long-run waiting time  $E[W]$ , at an M/G/1 system, so for server  $i$  we get

$$E[W_i] = \lambda_i E[S_i^2] \cdot \frac{1}{2(1 - \rho_i)} \quad \text{if } \rho_i < 1. \quad (3.8)$$

A type- $j$  job arrives in  $(t, t + dt)$  with probability  $\lambda p_j dt + o(dt)$ . With probability  $a_{ji}$  it is then assigned to server  $i$  where it experiences a total expected delay in system equal to  $E[W_i] + E[S_{ji}]$ . Thus the total expected delay for a job of type  $j$  is



$$E[D_j] = \sum_{i=1}^I a_{ji} (E[W_i] + E[S_{ji}]); \quad (3.9)$$

this becomes infinite if any  $\rho_i \geq 1$ . The total expected long-run *weighted* delay (per unit time) is

$$\begin{aligned} D(\underline{a}) &= \sum_{j=1}^J \lambda p_j d_j E[D_j] \\ &= \lambda \sum_{j=1}^J p_j d_j \left[ \sum_{i=1}^I a_{ji} (E[W_i] + E[S_{ji}]) \right]. \end{aligned} \quad (3.10)$$

From the perspective of jobs (or customers), it is natural to find a randomized job allocation that minimizes  $D(a)$  by solving the following nonlinear programming problem:

$$\text{P5:} \quad \min \sum_{j=1}^J p_j d_j \left[ \sum_{i=1}^I a_{ji} (E[W_i] + E[S_{ji}]) \right] \quad (3.11)$$

subject to constraints (3.2), (3.3), and (3.4)

where  $d_j$  is the weight for the delay of job  $j$ . When  $d_j = 1$ , the objective function of P5 is equivalent to minimizing the average delay of all  $J$  jobs. Alternately, the following problem minimizes the maximum expected delay:

$$\text{P6:} \quad \min \max_j \left\{ \sum_{i=1}^I a_{ji} (E[W_i] + E[S_{ji}]) \right\} \quad (3.12)$$

subject to constraints (3.2), (3.3), and (3.4).

### 3.3 Mean Non Linear Delay Penalty

Suppose it is important that the penalty for job delays be more stringent for long jobs, and in a manner that the (linear) long-run expectation of total delay does not reflect adequately. One such penalty parameterization is exponential:

exact penalty  $E\left[e^{\theta_i(W_i+S_{ji})}\right]$  where  $\theta_i > 0$ . The value of  $\theta_i$  is a decision maker's choice. Classical M/G/1 theory says that the limiting transform is

$$E\left[e^{\theta_i W_i}\right] = \frac{1 - \rho_i}{1 - \rho_i \left\{ \frac{E\left[e^{\theta_i S_i}\right] - 1}{\theta_i E[S_i]} \right\}} \quad (3.13)$$

provided the denominator is positive. To satisfy this requirement, the rate of input to server  $i$  must be, in general, smaller than what is allowed by the expected long-run waiting time formula in (3.8). Of course,

$$E\left[e^{\theta_i S_i}\right] = \sum_{j=1}^J p_j a_{ji} E\left[e^{\theta_i S_{ji}}\right] / \sum_{j=1}^J p_j a_{ji} \quad (3.14)$$

can quickly grow large, or become formally infinite, if any assignments of job type  $j'$  (denoting members of a subset of all jobs) are submitted to server  $i$ . Analogous to P5, the problem of minimize weighted penalty can be written as follows:

$$\text{P7} \quad \min \sum_{j=1}^J p_j d_j \left\{ \sum_{i=1}^I a_{ji} E\left[e^{\theta_i W_i}\right] E\left[e^{\theta_i S_{ji}}\right] \right\}$$

subject to constraints (3.2), (3.3), (3.4), and

$$0 < \rho_i \left\{ \frac{E\left[e^{\theta_i S_i}\right] - 1}{\theta_i E[S_i]} \right\} < 1 \quad \forall i = 1, 2, \dots, I. \quad (3.15)$$

The additional constraint is to ensure that the expression for  $E\left[e^{\theta_i W_i}\right]$  in the objective function is well defined. As before, when  $d_j = 1$ , P7 simply minimizes the average delay penalty for all  $J$  jobs. Similarly, the problem of minimizing the maximum penalty can be written as

$$P8 \quad \min \max_j \left\{ \sum_{i=1}^I a_{ji} E \left[ e^{\theta_i W_i} \right] E \left[ e^{\theta_i S_{ji}} \right] \right\}$$

subject to the constraints of (3.2), (3.3), (3.4), and (3.15).

#### 4. Incompatibility Models

Among all possible job to server assignments, it is convenient to stipulate that the assignment of job  $j$  to server  $j$  (or, equivalently, job  $i$  to server  $i$ ) is the most compatible, i.e.,  $E[S_{ii}] \leq E[S_{ji}]$  and  $E[S_{ii}^2] \leq E[S_{ji}^2]$  for  $j \neq i$ . Below are two incompatibility models that fit this stipulation.

##### 4.1 Proportionality Model

Suppose we represent incompatibility simply as follows: for  $k_{ji} \geq 1$ , we assume  $S_{ji} = k_{ji} S_{ii}$ , so

$$E[S_{ji}] = k_{ji} E[S_{ii}] \quad (4.1a)$$

$$E[S_{ji}^2] = k_{ji}^2 E[S_{ii}^2] \quad (4.1b)$$

for  $\forall j \neq i$ . Clearly the inequalities noted above hold, and the greater  $k$  is made the more flagrant is the incompatibility.

##### 4.2 Random Interruption Model

An alternative and physically plausible model is as follows. Let  $v_{ji}$  denote a Poisson rate of interruptions incurred by a job of type  $j$  when processed on server type  $i$ . These interruptions occur when the server must consult for advice, look for necessary materials, or rectify a breakdown that occurs. It is assumed here that the job is not displaced from the server's attention (set aside) during the interruption, so no other jobs may be done during the interruption. Analysis of a set-aside option will be conducted later. Let  $I_{ji}$  denote the random duration of an

interruption; successive interruptions are independent. Let  $S_{ji}^\#$  be the effective service time for server  $i$  to complete a job of type  $j$ , including the interruptions that occur. Then

$$S_{ji}^\# = S_{ii} + \sum_{\ell=1}^{N(S_{ii})} I_{ji}(\ell) \quad (4.2)$$

where, given  $S_{ii}$ ,  $N(S_{ii})$  is Poisson  $(v_{ji}S_{ii})$ .

By straightforward conditioning we find

$$E[S_{ji}^\#] = E[S_{ii}](1 + v_{ji}E[I_{ji}]) \quad (4.3)$$

$$E[(S_{ji}^\#)^2] = E[S_{ii}^2](1 + v_{ji}E[I_{ji}])^2 + v_{ji}E[S_{ii}]E[I_{ji}^2] \quad (4.4)$$

## 5. Numerical Examples

The problems presented above, P1, P2, P4, P5, P6, P7, and P8 were implemented and solved using the General Algebraic Modeling System (GAMS) developed by Brooke, Kendrick and Meeraus (1992). (Recall that P4 is linear version of P3.) Among all the constraints in these problems, constraints (3.2) and (3.15) cannot be implemented on a finite precision computer. Our implementation replaces equations (3.2) and (3.15) by the following:

$$\lambda \sum_{j=1}^J p_j a_{ji} E[S_{ji}] \leq 0.99 \quad \forall i = 1, \dots, I. \quad (5.1)$$

$$.01 \leq \rho_i \left\{ \frac{E[e^{\theta_i S_i}] - 1}{\theta_i E[S_i]} \right\} \leq .99. \quad (5.2)$$

For P1, P2, P4, P5, and P6, our example assumes that the first and second moments for service time are determined by the random interruption model.

Results for the proportionality model are similar. The data for our example are as follows:

- i)  $I = J = 6$ ,
- ii) The weight for the traffic intensity at server  $i$ ,  $c_i$ , equals 1 for all  $i$ ,
- iii) The cost of delay for job  $j$ ,  $d_j$ , equals 1 for all  $j$ ,
- iv) The probability of an incoming job being of type  $j$ ,  $p_j$ , is as follows:

Job	1	2	3	4	5	6
$p_j$	0.29	0.24	0.19	0.14	0.10	0.04

- v)  $v_{ji} = \text{Round}(U(1,4))$ , where  $U(a, b)$  is a Uniform random variable with parameters  $a$  and  $b$ , and  $\text{Round}()$  rounds a number to the nearest integer.
- vi)  $E[I_{ji}] = U(0.5, 1.5)$  and  $E[I_{ji}^2] = 2E[I_{ji}]^2$ .
- vii)  $E[S_{ii}] = U(0.5, 1.0)$  and  $E[S_{ii}^2] = 2E[S_{ii}]^2$ .

Table 1 shows the resulting matrices for the first and second moments generated based on the data from v, vi and vii.

**Table 1:** The first and second moments of the time server  $i$  takes to perform job  $j$ .

		$E[S_{ji}]$					
	Server1	Server2	Server3	Server4	Server5	Server6	
Job1	0.5364	1.3056	3.8214	2.6083	2.2060	1.3083	
Job2	1.3794	0.5878	2.4318	5.1551	1.2537	2.0121	
Job3	2.4959	1.6771	0.7628	3.7252	1.5549	0.9507	
Job4	1.6340	1.2613	2.3811	0.8751	2.8333	1.3421	
Job5	1.7820	2.0643	1.6229	1.5611	0.5891	1.4267	
Job6	1.5059	1.9296	3.1550	2.2705	2.1116	0.5171	

		$E[S_{ji}^2]$					
	Server1	Server2	Server3	Server4	Server5	Server6	
Job1	0.5754	4.2853	35.3372	15.8944	14.1712	4.6341	
Job2	5.1302	0.6911	15.4788	63.6172	4.6431	10.9792	
Job3	16.0385	6.9712	1.1638	32.3951	5.8913	2.5347	
Job4	6.8372	4.7252	14.7727	1.5316	21.7556	4.9190	
Job5	9.2434	12.2305	7.2066	5.9493	0.6940	5.1381	
Job6	5.7039	10.5089	24.9085	11.7939	12.8533	0.5347	



Using the above data, the maximum job arrival rate,  $\lambda_{\max}$ , can be obtained by solving the following linear programming problem:

P9:  $\lambda_{\max} = \max \lambda$   
*subject to constraints (3.3), (3.4), and*

$$\lambda \sum_{j=1}^I p_j a_{ji} E[S_{ji}] \leq 1 \quad \forall i = 1, \dots, I. \quad (5.3)$$

This problem yields  $\lambda_{\max} = 8.23$ . Corresponding to this  $\lambda_{\max}$ , there is also a feasible allocation,  $\hat{a}_{ji}$ , such that

$$\lambda_{\max} \sum_{j=1}^I p_j \hat{a}_{ji} E[S_{ji}] = 1 \quad \forall i = 1, \dots, I. \quad (5.4)$$

Table 2 summarizes the results for three different arrival rates: low ( $0.75\lambda_{\max}$ ), medium ( $0.85\lambda_{\max}$ ) and high ( $0.95\lambda_{\max}$ ). The first two allocations minimize the sum (P1) and the squared traffic intensities (P2) yield higher delays than the other three allocations. On the other hand, P1 and P2 produce smaller traffic intensities on the average. However, the maximum intensity for P1 and P2 for all three arrival rates are close to 1. Except for the low arrival rate, the maximum traffic intensity for both P1 and P2 is 0.99, the maximum allowed by constraint (5.1). This may not be desirable in practice.

The last three allocations minimize the maximum traffic intensity (P4), weighted delays (P5) and maximum delay (P6), have similar average intensity. Since minimizing the maximum traffic intensity does not directly take into account the delay, P4 generates slightly higher job delays than the other two. Observe that the traffic intensity for P4 is the same for every server. Similarly, the expected delay for every job is the same for problem P6.



Table 2: Summary of results from various optimization problems

Low Arrival Rate: $\lambda = 0.75\lambda_{\max}$					
	P1	P2	P4	P5	P6
	Minimize Sum Traffic	Minimize Sq. Traffic	Minimize Max Traffic	Minimize Weighted Delay	Minimize Max Delay
MIN E[Delay]	0.5927	1.0592	2.7176	2.0935	3.0698
AVE E[Delay]	5.0440	4.2503	3.6594	2.9473	3.0698
MAX E[Delay]	13.3894	13.3894	5.0153	3.5874	3.0698
MIN INTENS	0.1276	0.4002	0.7500	0.5850	0.6353
AVE INTENS	0.6620	0.6774	0.7500	0.7218	0.7304
MAX INTENS	0.9599	0.9599	0.7500	0.8183	0.8170
Medium Arrival Rate: $\lambda = 0.85\lambda_{\max}$					
	P1	P2	P4	P5	P6
	Minimize Sum Traffic	Minimize Sq. Traffic	Minimize Max Traffic	Minimize Weighted Delay	Minimize Max Delay
MIN E[Delay]	1.0018	1.1739	4.5287	3.8327	5.2134
AVE E[Delay]	31.8809	13.4553	6.2890	5.1513	5.2134
MAX E[Delay]	74.5428	49.1424	9.0138	6.1985	5.2134
MIN INTENS	0.4097	0.4535	0.8500	0.7383	0.7608
AVE INTENS	0.7748	0.7861	0.8500	0.8316	0.8357
MAX INTENS	0.9900	0.9900	0.8500	0.8930	0.8936
High Arrival Rate: $\lambda = 0.95\lambda_{\max}$					
	P1	P2	P4	P5	P6
	Minimize Sum Traffic	Minimize Sq. Traffic	Minimize Max Traffic	Minimize Weighted Delay	Minimize Max Delay
MIN E[Delay]	2.9863	2.9863	13.5844	12.2416	16.0495
AVE E[Delay]	33.9351	33.9351	19.4372	16.2206	16.0495
MAX E[Delay]	68.2769	68.2769	29.0061	20.6084	16.0495
MIN INTENS	0.7010	0.7010	0.9500	0.9118	0.9124
AVE INTENS	0.9199	0.9199	0.9500	0.9437	0.9442
MAX INTENS	0.9900	0.9900	0.9500	0.9640	0.9661

It is interesting to note that problems P5 and P6 which minimize functions of job delays produce allocations with traffic intensities similar to those generated by P4 which minimizes the maximum intensity. This similarity is probably due to the fact that, from (3.8), lower traffic intensity implies shorter waiting time which, in turn, implies shorter delay via (3.9). Moreover, this similarity illustrates that planning from the two perspectives, servers or jobs (customers), does not have to always be conflicting.

Problem P4, minimizing the maximum traffic intensity, tends to equalize the intensity at all servers. In Table 2, the minimum, average, and maximum values of traffic intensity for P4 all equal  $\alpha$ , if  $\lambda = \alpha\lambda_{\max}$ . In fact, the solution to problem P9, i.e.,  $\hat{a}_{ji}$ , also solves P4 because  $\hat{a}_{ji}$  is a feasible allocation and multiplying (5.3) by  $\alpha$  yields the following

$$\alpha\lambda_{\max} \sum_{j=1}^J p_j \hat{a}_{ji} E[S_{ji}] = \alpha \quad \forall i = 1, \dots, I$$

$$\lambda \sum_{j=1}^J p_j \hat{a}_{ji} E[S_{ji}] = \alpha \quad \forall i = 1, \dots, I.$$

So, the allocation  $\hat{a}_{ji}$  produces the same traffic intensity,  $\alpha$ , at all servers. Since  $\alpha$  is the maximum intensity,  $\hat{a}_{ji}$  must be optimal to P4 also.

Unlike the other problems, the distribution for service times must be specified for problems P7 and P8. To simplify our illustration, replace  $v$  to  $v_{ii}$  with the following:

- viii) The service time  $S_{ii}$  has a gamma distribution with parameters  $\alpha_i$  and  $\beta_i$ , and

$$E[S_{ii}] = \beta_i / \alpha_i \quad \text{and} \quad E[S_{ii}^2] = (\beta_i^2 + \beta_i) / \alpha_i^2, \text{ and}$$

- ix) The service time  $S_{ji}$  has a gamma distribution with parameters  $\alpha_i$  and  $k_{ji}\beta_i$ , and

$$E[S_{ji}] = k_{ji}\beta_i / \alpha_i \quad \text{and} \quad E[S_{ji}^2] = \left( (k_{ji}\beta_i)^2 + k_{ji}\beta_i \right) / \alpha_i^2.$$

The above assumptions is similar to the proportionality model, in that equation (4.1a) holds. However, equation (4.1b) does not and the following holds instead:

$$E[S_{ji}^2] < k_{ji}^2 E[S_{ii}^2] \quad \forall j \neq i.$$

Moreover,

$$E\left[e^{\theta_i S_{ji}}\right] = \left(1 - \frac{\theta_i \text{Var}[S_{ji}]}{E[S_{ji}]}\right)^{-E[S_{ji}]^2 / \text{Var}[S_{ji}]} \quad \forall i \text{ and } j.$$

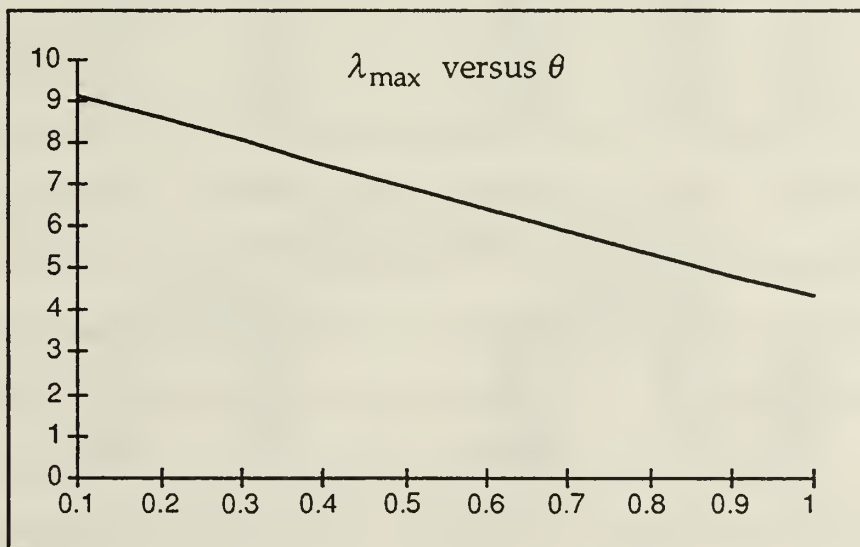
For our example,

- 1)  $\alpha_i = 1$  and  $\beta_i = 2 \quad \forall i$
- 2)  $k_{ji} = \frac{|j-i|}{2} + 1, \quad \forall j \neq i.$

Determining the value of  $\lambda_{\max}$  now requires solving problem P9 with the addition of the following constraint:

$$0 \leq \rho_i \left\{ \frac{E[e^{\theta_i S_i}] - 1}{\theta_i E[S_i]} \right\} \leq 1, \quad \forall i. \quad (5.5)$$

Since (5.5) depends on  $\theta$ , improper choice for  $\theta$  may render problems P7 and P8 infeasible. Figure 1 below displays a graph of  $\lambda_{\max}$  as a function of  $\theta$  under the assumption that  $\theta_i = \theta$  for all  $i$ . So, the choice of  $\theta$  limits the application of



**Figure 1:** A graph of  $\lambda_{\max}$  as a function of  $\theta$  under the assumption that  $\theta_i = \theta$  for all  $i$ .

problem P7 and P8. Certainly, when the arrival rate is 0.7, it is not possible to choose  $\theta$  to be 1 and vice versa. Tables 3 and 4 compare the results with  $\theta = 0.7$  for two pairs of problems, P5 & P7 and P6 & P8, respectively. In Table 3, minimizing weighted job delays (P5) and job penalties (P7) generate similar job delays for all three arrival rates. On the other hand, only the job penalties for low arrival rate are similar. At the high arrival rate, the infinite penalties for jobs 1 and 6 under P5 indicate that the penalty function is undefined, i.e., constraint (3.15) is not satisfied. Problems P6 and P8 display a similar behavior in Table 4.

Table 3: Comparing job delays and penalties for problems P5 and P7.

Low Arrival Rate: $\lambda = 0.75\lambda_{\max}$					
	P7: Minimize Weighted Penalty		P5: Minimize Weighted Delay		
	DELAY	PENALTY	DELAY	PENALTY	
Job 1	1.1957	4.3649	1.2562	6.0654	
Job 2	1.0157	3.4013	1.0504	3.7571	
Job 3	0.9296	2.8850	0.8663	2.5424	
Job 4	0.8693	2.7167	0.7537	2.1365	
Job 5	0.8134	2.5799	0.7008	2.0217	
Job 6	0.7664	2.6051	0.7668	2.6076	
Ave	1.0008	3.3728	0.9873	3.7493	
Medium Arrival Rate: $\lambda = 0.85\lambda_{\max}$					
	P7: Minimize Weighted Penalty		P5: Minimize Weighted Delay		
	DELAY	PENALTY	DELAY	PENALTY	
Job 1	1.3395	6.4635	1.4309	14.1940	
Job 2	1.1525	4.9695	1.1788	5.4984	
Job 3	1.0767	4.1320	0.9700	3.1213	
Job 4	1.0059	3.8486	0.8595	2.5732	
Job 5	0.9456	3.7056	0.8320	2.6513	
Job 6	0.8858	3.8450	0.9493	5.1916	
Ave	1.1404	4.9153	1.1237	6.8620	
High Arrival Rate: $\lambda = 0.95\lambda_{\max}$					
	P7: Minimize Weighted Penalty		P5: Minimize Weighted Delay		
	DELAY	PENALTY	DELAY	PENALTY	
Job 1	1.5106	15.5540	1.6392	$\infty$	
Job 2	1.3238	11.5910	1.3401	14.0168	
Job 3	1.2673	9.3830	1.1036	4.4200	
Job 4	1.1917	8.8212	0.9955	3.4966	
Job 5	1.1185	8.6550	0.9977	4.3668	
Job 6	1.0326	9.2758	1.1772	$\infty$	
Ave	1.3166	11.5468	1.2929	N/A	

Table 4: Comparing job delays and penalties for problems P6 and P8.

Low Arrival Rate: $\lambda = 0.75\lambda_{\max}$					
	P8: Minimize Maximum Penalty		P6: Minimize Maximum Delay		
	DELAY	PENALTY	DELAY	PENALTY	
Job 1	1.1188	3.7831	1.0998	3.8200	
Job 2	1.0730	3.7831	1.0998	3.8304	
Job 3	1.0803	3.7831	1.0998	3.7653	
Job 4	1.0649	3.7831	1.0998	4.0381	
Job 5	1.0209	3.7831	1.0998	4.0422	
Job 6	0.8637	3.0756	1.0025	3.8652	
Ave	1.0729	3.7548	1.0959	3.8667	
Medium Arrival Rate: $\lambda = 0.85\lambda_{\max}$					
	P8: Minimize Maximum Penalty		P6: Minimize Maximum Delay		
	DELAY	PENALTY	DELAY	PENALTY	
Job 1	1.2847	5.6349	1.2438	5.4339	
Job 2	1.2045	5.6349	1.2438	6.3660	
Job 3	1.1525	5.6349	1.2438	8.7200	
Job 4	1.0736	4.9570	1.2438	12.7694	
Job 5	1.1845	5.6349	1.2438	13.7092	
Job 6	1.0599	5.2289	1.2438	21.1060	
Ave	1.1918	5.5237	1.2438	8.7634	
High Arrival Rate: $\lambda = 0.95\lambda_{\max}$					
	P8: Minimize Maximum Penalty		P6: Minimize Maximum Delay		
	DELAY	PENALTY	DELAY	PENALTY	
Job 1	1.4910	13.1304	1.4212	15.6351	
Job 2	1.3354	13.1304	1.4212	6.7326	
Job 3	1.3079	13.1304	1.4212	15.5092	
Job 4	1.2246	11.9743	1.4212	1.2211	
Job 5	1.1317	9.6943	1.4212	$\infty$	
Job 6	1.0162	7.9914	1.2959	$\infty$	
Ave	1.3267	12.4194	1.4162	N/A	

## 6. Application

One application of the optimization problems is in quantifying the benefit of additional training. In all of the above examples,  $p_6 = 0.04$ . So, there are not many jobs of type 6 in the system. Therefore, server 6, who is the expert for job type 6, has to process other types of jobs in order to, e.g., minimize the maximum expected delay among all 6 job types, (problem P6). In order to improve this performance measure, it is logical to train server 6 to become an expert at processing another job besides job 6. Assume that, if server 6 is trained to process job  $i$ , then



$$E[S_{6i}] = E[S_{ii}] \text{ and } E[S_{6i}^2] = E[S_{ii}^2].$$

By resolving problem P6 with the first two moments of its service times suitably modified, the training benefit for server 6 can be measured quantitatively. Using data from the interruption model above (i.e., i – vii) and high arrival rate, Figure 2 displays the maximum delay for each training alternative.

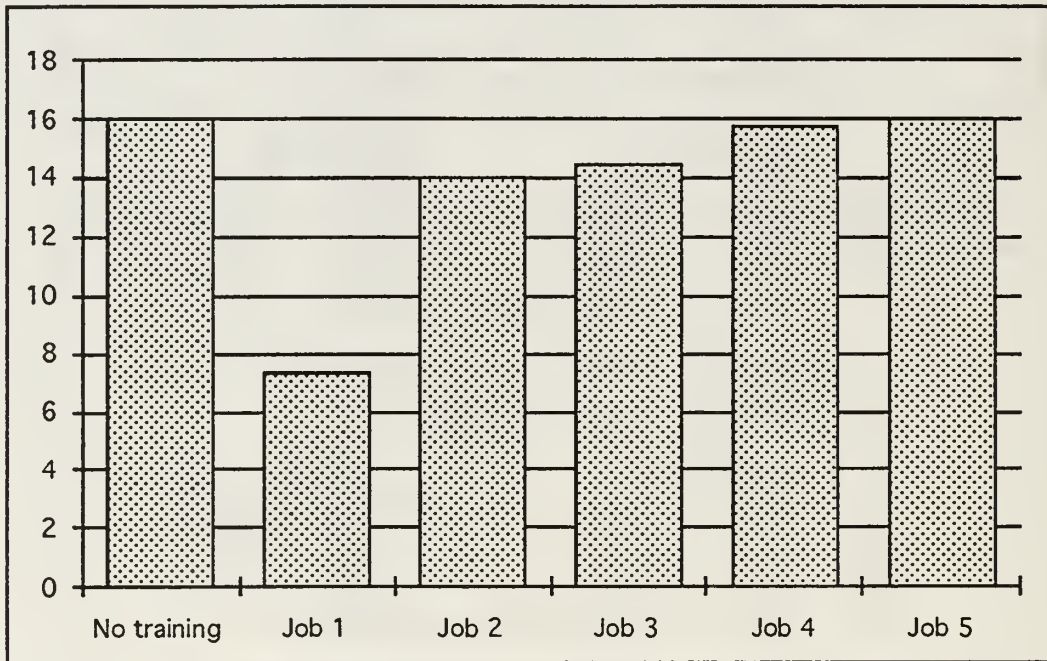


Figure 2: Maximum Job Delay vs. Training Alternatives for Server 6

As expected, it is best to train server 6 to process job 1, for doing so decreases the maximum job delay by the largest amount.

In general, it is also possible to formulate an optimization problem for assigning servers to training programs for a given budget constraint.



## REFERENCES

- M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, Inc., New York, New York (1993).
- A. Brooke, D. Kendrick, and A. Meeraus, *GAMS: A User's Guide*, Release 2.25, The Scientific Press, South San Francisco, California (1992).
- D.P. Gaver and P.A. Jacobs "Nonparametric estimation of the probability of a long delay in the M/G/1 queue," *J. Royal Statist. Soc. Ser B.*, 5 (1988), 392-402.
- L. Kleinrock *Queueing Systems, Volume I: Theory*, John Wiley & Sons, New York, NY 1975.



## DISTRIBUTION LIST

1. Research Office (Code 09) .....1  
Naval Postgraduate School  
Monterey, CA 93943-5000
2. Dudley Knox Library (Code 013) ..... 2  
Naval Postgraduate School  
Monterey, CA 93943-5002
3. Defense Technical Information Center ..... 2  
8725 John J. Kingman Rd., STE 0944  
Ft. Belvoir, VA 22060-6218
4. Therese Bilodeau ..... 1  
Dept of Operations Research  
Naval Postgraduate School  
Monterey, CA 93943-5000
5. Prof. Donald P. Gaver (Code OR/Gv)..... 5  
Naval Postgraduate School  
Monterey, CA 93943-5000
6. Prof. Patricia A. Jacobs (Code OR/Jc) ..... 5  
Naval Postgraduate School  
Monterey, CA 93943-5000
7. Prof. Siriphong Lawphongpanich (Code OR/Lp) ..... 5  
Naval Postgraduate School  
Monterey, CA 93943-5000
8. Mr. Kevin Becker ..... 1  
Tandem Computers  
3642 Springbrook Ave.  
San Jose, CA 95148
9. Dr. J. Abrahams ..... 1  
Code 111, Room 607  
Mathematical Sciences Division, Office of Naval Research  
800 North Quincy Street  
Arlington, VA 22217-5000
10. Dr. John C. Bailar ..... 1  
Department of Health Studies  
University of Chicago  
5841 S. Maryland Ave., MC 2007  
Chicago, IL 60637-1470

11. Prof. D. R. Barr ..... 1  
 Dept. of Systems Engineering  
 U.S. Military Academy  
 West Point, NY 10996
  
12. Dr. David Burman ..... 1  
 AT&T Bell Telephone Laboratories  
 600 Mountain Avenue  
 Murray Hill, NJ 07974
  
13. Center for Naval Analyses ..... 1  
 4401 Ford Avenue  
 Alexandria, VA 22302-0268
  
14. Dr. Edward G. Coffman, Jr..... 1  
 AT&T Bell Telephone Laboratories  
 600 Mountain Avenue  
 Murray Hill, NJ 07974
  
15. Prof. Sir David Cox ..... 1  
 Nuffield College  
 Oxford OX1 1NF  
 ENGLAND
  
16. Prof. H. G. Daellenbach ..... 1  
 Dept. of Operations Research  
 University of Canterbury  
 Christchurch  
 NEW ZEALAND
  
17. Dr. D. F. Daley ..... 1  
 Statistics Dept. (I.A.S.)  
 Australian National University  
 Canberra, A.C.T 2606  
 AUSTRALIA
  
18. Mr. Mike Davis ..... 1  
 R55  
 9800 Savage Road  
 Ft. Meade, MD 20755
  
19. Dr. B. Doshi..... 1  
 AT&T Bell Laboratories  
 HO 3M-335  
 Holmdel, NJ 07733

20.	Dr. Naihua Duan .....	1
	RAND Corporation	
	Santa Monica, CA 90406	
21.	Dr. Guy Fayolle.....	1
	I.N.R.I.A.	
	Dom de Voluceau-Rocquencourt	
	78150 Le Chesnay Cedex	
	FRANCE	
22.	Dr. M. J. Fischer.....	1
	Defense Communications Agency	
	1860 Wiehle Avenue	
	Reston, VA 22070	
23.	Prof. George S. Fishman .....	1
	Curr. in OR & Systems Analysis	
	University of North Carolina	
	Chapel Hill, NC 20742	
24.	Dr. Neil Gerr.....	1
	Office of Naval Research	
	Arlington, VA 22217	
25.	Dr. R. J. Gibbens .....	1
	Statistics Laboratory	
	16 Mill Lane	
	Cambridge	
	ENGLAND	
26.	Prof. Peter Glynn .....	1
	Dept. of Operations Research	
	Stanford University	
	Stanford, CA 94305	
27.	Prof. Linda V. Green .....	1
	Graduate School of Business	
	Columbia University	
	New York, NY 10027	
28.	Dr. Shlomo Halfin.....	1
	Bellcore	
	445 South Street	
	Morristown, NJ 07962-1910	
	(MRE 2L309)	

29. Prof. J. Michael Harrison ..... 1  
 Graduate School of Business  
 Stanford University  
 Stanford, CA 94305-5015
30. Dr. P. Heidelberger ..... 1  
 IBM Research Laboratory  
 Yorktown Heights  
 New York, NY 10598
31. Prof. D. L. Iglehart ..... 1  
 Dept. of Operations Research  
 Stanford University  
 Stanford, CA 94305-5015
32. Institute for Defense Analysis ..... 1  
 1800 North Beauregard  
 Alexandria, VA 22311
33. Dr. F. P. Kelly ..... 1  
 Statistics Laboratory  
 16 Mill Lane  
 Cambridge  
 ENGLAND
34. LCDR Paul Knechtges ..... 1  
 Program Manager, Fleet Occupational Health  
 Naval Medical Research & Development Command  
 Bethesda, MD 20889-5606
35. Mr. Koh Peng Kong ..... 1  
 OA Branch, DSO  
 Ministry of Defense  
 Blk 29 Middlesex Road  
 SINGAPORE 1024
36. Prof. Guy Latouche ..... 1  
 University Libre Bruxelles  
 C.P. 212, Blvd. De Triomphe  
 Bruxelles B-1050  
 BELGIUM
37. Dr. A. J. Lawrance ..... 1  
 Dept. of Mathematics  
 University of Birmingham  
 P.O. Box 363  
 Birmingham B15 2TT  
 ENGLAND



38. Prof. J. Lehoczky ..... 1  
 Department of Statistics  
 Carnegie-Mellon University  
 Pittsburgh, PA 15213
39. CAPT (Sel) Mike Lilienthal ..... 1  
 Aviation Medicine/Human Performance Research  
 Naval Medical Research & Development Command  
 Bethesda, MD 20889-5044
40. Dr. D. M. Lucantoni ..... 1  
 AT&T Bell Laboratories  
 Holmdel, NJ 07733
41. Dr. James R. Maar ..... 1  
 National Security Agency  
 9608 Basket Ring  
 Columbia, MD 21045-0689
42. Dr. Colin Mallows ..... 1  
 AT&T Bell Telephone Laboratories  
 600 Mountain Avenue  
 Murray Hill, NJ 07974
43. Prof. M. Mazumdar ..... 1  
 Dept. of Industrial Engineering  
 University of Pittsburgh  
 Pittsburgh, PA 15235
44. Dr. James McKenna ..... 1  
 Bell Communications Research  
 445 South Street  
 Morristown, NJ 07960-1910
45. Prof. Paul Moose ..... 1  
 C3I Academic Group  
 Naval Postgraduate School  
 Monterey, CA 93943-5000
46. Dr. John A. Morrison ..... 1  
 AT&T Bell Telephone Laboratories  
 600 Mountain Avenue  
 Murray Hill, NJ 07974

47.	Dr. T. J. Ott..... Bellcore 445 South Street Morristown, NJ 07962-1910 (MRE 2P388)	1
48.	Dr. V. Ramaswami..... MRE 2Q-358 Bell Communications Research Inc. 445 South Street Morristown, NJ 07960	1
49.	Dr. Martin Reiman..... Rm #2C-117 AT&T Bell Laboratories 600 Mountain Avenue Murray Hill, NJ 07974-2040	1
50.	Prof. Maria Rieders..... Dept. of Industrial Engineering Northwestern University Evanston, IL 60208	1
51.	Dr. Rhonda Righter ..... Dept. of Decision & Info. Sciences Santa Clara University Santa Clara, CA 95118	1
52.	Prof. G. Shanthikumar ..... The Management Science Group School of Business Administration University of California Berkeley, CA 94720	1
53.	Prof. N. D. Singpurwalla ..... George Washington University Washington, DC 20052	1
54.	Prof. H. Solomon..... Department of Statistics Sequoia Hall Stanford University Stanford, CA 94305	1

55. Prof. L. C. Thomas ..... 1  
 Dept. of Business Studies  
 William Robertson Building  
 50 George Square  
 Edinburgh, EH8 9JY  
 SCOTLAND
56. Dr. D. Vere-Jones ..... 1  
 Dept. of Math  
 Victoria Univ. of Wellington  
 P.O. Box 196  
 Wellington  
 NEW ZEALAND
57. Mr. Lai Kah Wah ..... 1  
 Operations Analysis Department  
 Systems & Computer Organisation  
 19th Storey, Defence Technology Tower B  
 Depot Road 0410  
 SINGAPORE
58. Dr. Ed Wegman ..... 1  
 George Mason University  
 Fairfax, VA 22030
59. Dr. L. Wein ..... 1  
 Operations Research Center, Rm E40-164  
 Massachusetts Institute of Technology  
 Cambridge, MA 02139
60. Dr. Alan Weiss ..... 1  
 Rm 2C-118  
 AT&T Bell Laboratories  
 600 Mountain Avenue  
 Murray Hill, NJ 07974-2040
61. Dr. P. Welch ..... 1  
 IBM Research Laboratory  
 Yorktown Heights, NY 10598
62. Prof. Roy Welsch ..... 1  
 Sloan School  
 M.I.T.  
 Cambridge, MA 02139







DUDLEY KNOX LIBRARY



3 2768 00325735 3